CLUBE Z81

NUMERO TRES -30/DEZEMBRO/1982 .
COORDENADORES : maria irene E alberto fernandes
AV.BOAVISTA-832,2.T 4100 PORTO

NESTE NÚMERO

PRIMEIRA PÁGINA

Chegou-nos uma carta de Vila Velha de Ródão, do Sr. António Rodrigues, com o seguinte comentário:

"Para o preço da inscrição, achei um pouco pobre (o boletim)".

Conforme temos solicitado, as críticas vem chegando e esta é mais uma que consideramos oportuna pois leva-nos a um esclarecimento que deve ser divulgado. Aparentemenmos oportuna pois leva-nos a um esclarecimento que deve ser divulgado. Aparentementemente, o valor da inscrição pode parecer exagerado, mas vamos a contas:

	• Ordenado mensal da coordenadora (80 horas/mês dedicadas ao Clube)	7 000$00
105 inscritos	• Despesas de fotocópias dos boletins (105 boletins × 30$00)	3 150$00
	• Despesas de correio p/ enviar boletins (105 × 11$00)	1 155$00
	• Aluguer de instalações	-
	• Telefone	-
	Total de despesas mensais	11 305$00
	" " " anuais (11 305$00 ×12)	135 660$00

Se pensarmos que o Clube se resume apenas ao boletim, um ano de cotizações dos seus membros equivale a (1 500$00 × 12) **157 500$00**

• N/ saldo anual (157 500$00 – 135 660$00) **21 840$00**

Será que um saldo anual de 22 000$oo é suficiente para despesas de aluguer, telefone e outras (todos os dias respondemos individualmente a cartas recebidas) ?

Não duvidamos que a maioria dos membros do Clube se associou ao mesmo para a obtenção do boletim. Se fosse essa apenas a finalidade do Clube, a ideia seria bastante pobre. O Clube existe para tentar resolver os pequenos problemas dos utilizadores; para lançar ideias novas ou antigas; para trocas de experiencias ou programas. Gostaríamos de transmitir esta ideia e, fundamentalmente, esperamos que as críticas cheguem acompanhadas das soluções. Os boletins e tudo o mais que fizermos só deixarão de ser pobres desde que todos colaborem para os aperfeiçoarem.

Temos conhecimento de que a "Landry" vai lançar uma publicação comercial a exemplo do que já acontece com a "Cérebro" e a "Digitus".

Será que os membros do Clube querem transformar a nossa publicação num jornal com publicidade e distribuição nas livrarias?

Já temos pedidos nesse sentido – gostaríamos de conhecer a vossa opinião individual.

Maria Irene O. Santos

Alberto C. Fernandes

PASSO A PASSO

TÉCNICAS DE PROGRAMAÇÃO: O PROCESSO DE ORDENAÇÃO

Eu não sabia exactamente como funcionava o método de ordenação ("bubble sorts"). Então decidi investigar a fundo se seria possível idealizar programas que tornassem esse processo auto-explanatório, e depois verificar como isso acontecia.
Quando comecei, fiquei completamente baralhado. Agora, penso que já compreendi.
Apesar do método de ordenação não ser particularmente rápido ou sofisticado, é suficientemente adequado para pequenas tabelas, com a vantagem de ocupar um pequeno espaço da memória.

ORDENAÇÃO RÁPIDA

Para começar a compreender o método de ordenação, dê entrada ao programa que apresentamos na fig. 1. Ao fazê-lo, observe atentamente cada linha. O programa permite-lhe introduzir oito números à sua escolha e pela ordem que preferir. Depois o computador entra em modo FAST e, em poucos segundos, escreve-os por ordem crescente. Isto dá-lhe já uma pequena ideia da capacidade do método de ordenação.
Um exame da listagem da fig. 1 revelar-lhe-á o que se pretende que a máquina realize sucessivamente:

```
 2 DIM A(8)
10 FOR J=1 TO 8
15 INPUT B
16 LET A(J)=B
17 PRINT A(J)
20 NEXT J
50 FOR J=1 TO 8
52 FAST
55 LET K=J+1
60 FOR I=K TO 8
65 LET S=K+8-I
70 IF A(S)>A(J) OR A(S)=A(J)
   THEN GOTO 90
```

Fig. 1 →

```
75 LET M=A(S)
80 LET A(S)=A(J)
85 LET A(J)=M
90 NEXT I
95 NEXT J
96 PRINT
97 FOR J=1 TO 8
98 PRINT A(J)
99 NEXT J
```

Fig. 1 (continuação)

"Considere o número que se encontra no início da lista e compare-o com cada um dos números dessa lista a partir do que se encontra na última posição até encontrar um que lhe seja inferior; depois troque a posição desses dois números (se não for detectado um número menor, re-inicie o processo com o número que a seguir se encontra na lista)".

Deste modo, os números menores vão subindo para o topo da lista, ao passo que os maiores vão descendo. Sirva-se de papel e lápis e escreva oito números desordenadamente. Depois utilize as instruções dadas ao computador, re-escrevendo a lista após cada troca de números. Poderá verificar que por este simples processo consegue, na realidade, ordenar a lista por ordem crescente.

ORDENAÇÃO LENTA

Uma maneira simples de verificar o procedimento do computador é abandonar a exigência de rapidez e, deliberadamente, abrandar a marcha do programa, obrigando-o a indicar cada par de números cuja posição na lista irá trocar.
Observe o programa da figura 2* e execute-o. O computador exibe duas colunas de números: a da esquerda é a da lista desordenada inicial; a da direita vai indicando as sucessivas modificações de ordenação, mostrando (com um pequeno quadrado preto) os pares que vão sendo trocados.

* V. pág. seguinte

→

```
 2 DIM A(8)
 5 LET Y=0
10 FOR J=1 TO 8
15 LET A(J)=INT  (AND*89+10)
20 NEXT J
21 LET X=0
22 FOR J=1 TO 8
23 FOR K=1 TO 20
24 NEXT K
25 PRINT AT X,Y;A(J)
28 LET X=X+2
30 NEXT J
40 LET Y=Y+3
50 FOR J=1 TO 8
55 LET K=J+1
60 FOR I=K TO 8
65 LET S=K+ 8-I
70 IF A(S) > A(J)  OR A(S)=A(J)  THEN
   GOTO 90
75 LET M=A(S)
80 LET A(S)=A(J)
85 LET A(J)=M
86 PRINT AT 2*S-2,Y-1;"▌"
87 PRINT AT 2*J-2,Y-1;"▌"
89 GOTO 21
90 NEXT I
95 NEXT J
96 PRINT "SORTED"
```

Fig. 2

Quando o processo tiver terminado, a máqui na mostra-lhe uma figura semelhante à da figura 3.

```
94 25
33 26
55 29
29 33
49 35
35 49
25 55
26 94
SORTED
```

Fig. 3

Com algumas modificações no programa da fig. 2, obtém-se o que está indicado na fig. 4. Este obriga ao registo de cada permutação feita, até que a ordenação se complete (fig. 5).

Uma observação final que consideramos de interesse é a de que o número de permuta ções necessárias é sempre aproximadamente igual ao número de elementos da lista a ordenar.

```
 2 DIM A(8)
 5 LET Y=11
10 FOR J=1 TO 6
15 LET A(J)=INT  (RND*89+10)
20 NEXT J
21 LET X=0
22 FOR J=1 TO 6
23 FOR K=1 TO 20
24 NEXT K
25 PRINT AT X,Y;" ";A(J)
26 LET X=X+2
30 NEXT J
40 LET Y=14
50 FOR J=1 TO 6
55 LET K=J+1
60 FOR I=K TO 8
65 LET S=K+8-I
70 IF A(S) > A(J)  OR  A(S)=A(J)  THEN
   GOTO 90
75 LET M=A(S)
80 LET A(S)=A(J)
85 LET A(J)=M
86 PRINT AT 2*S-2,14;"▌"
87 PRINT AT 2*J-2,14;"▌"
89 GOTO 21
90 NEXT I
95 NEXT J
96 PRINT "SORTED"
```

Fig. 4

```
50▌26▌12 12 12 12 12
76 76 76▌50▌26 26 26
12 12▌26 26▌50 50 50
71 71 71 71 71 71 71
86 86 86 86 86▌76 76
26▌50 50▌76 76▌86 86
98 98 98 98 98 98▌86
86 86 86 86 86 86▌98
SORTED
```

Fig. 5

ESTRUTURAS DE DADOS: OS ALIMENTOS DA PROGRAMAÇÃO

Os dados estão para a informática como os ingredientes para a culinária - são a base de qualquer realização.
Mas esses dados, as informações que tratam um programa, não podem apresentar-se simplesmente em bloco. Tem que ser classificados, endereçados, manipulados.

Há uma estrutura que preside à sua organização, que comporta, como qualquer unidade informática, um duplo aspecto lógico e físico.
O primeiro dá origem à organização abstracta, desligada dos aspectos materiais.
O segundo proporciona a sua realização concreta, sem a qual não pode existir abstracção.
É nesta dupla óptica que examinaremos algumas das estruturas de dados mais utilizadas actualmente.

A programação estruturada equivale à divisão metódica e hierárquica dos programas em sub-programas. Organizar as acções é uma etapa inicial, mas insuficiente desde que não haja uma abordagem àquilo que se considera a vida da programação: os dados.
Depois de se descreverem os tipos de dados base, veremos como é possível tratar estruturas mais complexas.
A programação estruturada passa, efectivamente, pela organização dos dados e, quer se trate de acções ou de informações, o processo é identico: modularidade, análise lógica, implantação física são alguns dos aspectos que também aqui encontraremos.
Mas, antes de mais, o que é um dado? Primeiro que tudo, é uma informação: 3, 4, Ø, saldo da conta bancária, texto de um livro... são dados tratáveis pela informática.
Os dados estão reagrupados em classes, nas quais todos os elementos possuem a mesma propriedade: por exemplo, os números podem ser adicionados, diminuídos, multiplicados, etc.; um texto pode ser completado pela inserção de outro, algumas palavras podem ser modificadas, etc.
Cada classe possui, no entanto, características operatórias específicas. É aí que reside a noção de estrutura de dados: definir um tipo de dado implica descrever o conjunto das operações realizáveis para cada elemento pertencente a esse tipo. Por isso, fala-se de tipo abstracto para os diferenciar da sua realização interna. Dado que os computadores, ao nível físico, não manipulam abstracções mas octetos, a arte da informática consiste em libertar-se destes aspectos e atingir níveis de abstracção cada vez mais elevados. O desenvolvimento das linguagens LOGO, APL ou ADA provam isso.
O programador deverá ter conciencia das limitações da linguagem que usa, de modo a utilizar conceitos mais generalizados que lhe sejam úteis como instrumentos intelectuais e lhe permitam realizar lógicas mais funcionais, mais próximas do objectivo a atingir.
No quadro das estruturas de dados, distinguem-se dois níveis de descrição: o nível lógico ligado à descrição das operações permitidas e às suas propriedades, e o nível físico que corresponde à técnica e ao modo de implantação das estruturas lógicas no computador, atendendo aos aspectos "hard" ou "soft" (físico ou lógico).
Cada linguagem tem sempre as suas inclinações específicas: o BASIC, quadros e séries de caracteres; o PASCAL, ponteiros e registos; o LISP, listas, etc. Isso não significa que estes aspectos físicos derivados dos compiladores e dos interpretadores, devam ser considerados inibitórios.

→

Na realidade, é possível - utilizando a abordagem funcional que é nosso propósito - implantar várias listas em BASIC e matrizes em LISP, por exemplo. Programas de Inteligencia Artificial, que necessitam de estruturas mais sofisticadas (arborescencias, grafos, redes semanticas, etc.), foram no entanto escritos em FORTRAN que conhece apenas os quadros numéricos e algumas manipulações de séries de caracteres. É portanto indispensável, para quem quiser escrever programas mais interessantes, conhecer as características lógicas e as implantações físicas de algumas grandes famílias de estruturas de dados, antes de actuar no sentido de definir um novo tipo aquando da análise e concepção de uma lógica.

OS TIPOS DE BASE

Chama-se tipos de base aos tipos de dados elementares que permitem uma realização física simples e imediata. A maioria das linguagens de programação permite a sua utilização, graças a instruções previstas para esse efeito. Dados escalares e quadros de dados são geralmente considerados como a base de toda a programação.

OS DADOS ESCALARES

São os tipos de dados mais simples, cuja estrutura se reduz a um único elemento - por exemplo: 3, 4, "D", Verdadeiro.
Estes tipos de dados são frequentemente definidos na própria linguagem: números inteiros ou reais (simples ou dupla precisão), caracteres (de "0" a "9", de "a" a "z" e de "A" a "Z", mais os caracteres especiais "+", ":", "α", etc.), enfim, dados "booleanos" *. Por vezes - como em PASCAL ou ADA - é possível criar os seus próprios dados escalares, utilizando dois mecanismos diferentes: a enumeração, que equivale a descrever a totalidade dos valores possíveis que uma variável deste tipo pode assumir, ou o intervalo, que consiste na restrição da totalidade dos valores possíveis de um tipo pré-definido. Por exemplo, o tipo dia-semana será enumerado por: segunda-feira, terça, quarta, quinta, sexta, sábado, domingo; e o tipo número-dia-ano pelo intervalo 1...366, sub-conjunto dos números inteiros.
As operações possíveis com todos os dados escalares são:
a definição e a criação, por vezes implícitas em certas linguagens interpretadas (BASIC, APL e, em certos casos, FORTRAN), e explícitas noutras(PASCAL, ALGOL, C, ADA); a afectação e a leitura de um valor numa variável.
Certas operações estão limitadas a um tipo particular. A adição, a multiplicação, a subtracção e a divisão encontram-se em todos os tipos numéricos, com todas as comparações possíveis: igualdade, relações de ordem, etc. Os dados "booleanos" permitem as operações lógicas "e", "ou" e as suas combinações, a negação e o teste de igualdade.
Os caracteres apenas permitem a comparação.
Na figura 1 pode ver-se a representação física desses dados.

136098
V
1.3OE5
"A"

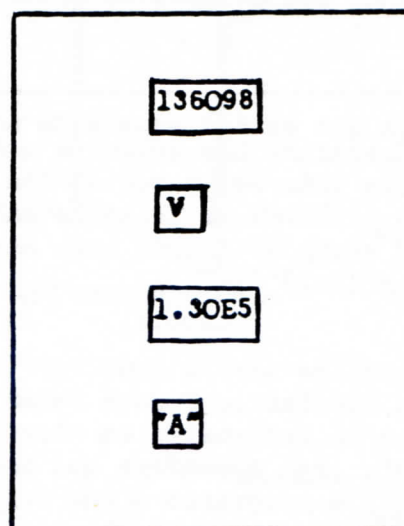



Fig. 1

Os dados escalares apenas ocupam uma única célula da memória e são considerados tipos de base.

Cada dado ocupa uma célula da memória: um octeto para os caracteres, dois ou mais para os inteiros, quatro ou mais para os reais, e um único bit é suficiente para memorizar as variáveis "booleanas".

(CONTINUA NOS PRÓXIMOS NÚMEROS)

*Termo derivado da álgebra de BOOLE

SECÇÃO DO LEITOR

.DÚVIDAS...SUGESTÕES...COMENTÁRIOS...OPINIÕES...DÚVIDAS...SUGESTÕES...COMENTÁRIOS...OPINI

"Tenho uma impressora GP-100 que gostaria de ligar ao ZX81. Ao ler o livro de instruções que a acompanha, fiquei quase na mesma, pois que o mesmo só fornece os códigos mas não exemplifica como programá-los." DOMINGOS MORAIS
Monte de Caparica

As impressoras com entrada tipo Paralelo/Centronics necessitam de um interface especial para ligar ao ZX81. O interface distribuído pelo importador da Seikosha está esgotado neste momento. Mas, o mesmo importador vai possuir brevemente um interface fabricado pela Memotech, que permitirá o uso de impressoras com papel comum, embora continuando a imprimir linhas com 32 caracteres e não respondendo aos caracteres gráficos do ZX81.

"Tenho um colega que está à espera de adquirir um ZX Spectrum. Ele acredita que o ZX Microdrive (não sei o que é) lhe vai resolver o problema da falta de RAM. Eu pergunto: É verdade? Existe esse Microdrive para o ZX81?" HUGO ASSUMPÇÃO
Lisboa

Existe um Microdrive para o Spectrum (já o vimos na Sinclair). Poderá reter até 100 K bytes de informação em Micro-diskettes de 3"; isso significa que pode guardar dados e programas a uma velocidade de 16 K/seg.
Para o ZX81 existem Microdrives, mas não da Sinclair. Mais do que uma das pequenas firmas que em Inglaterra trabalham em acessórios para o ZX81 anunciam Interfaces e Floppies para esta máquina. Em Portugal ainda ninguém importou este acessório (talvez porque o preço é elevado).

"Trabalho com um ZX81. Dentro de um ciclo FOR-NEXT faz-se a introdução de uma string A$ função de I, que portanto varia de cada vez que se completa um ciclo. Gostaria de saber como fazer para guardar todas essas strings, para que seja possível trabalhar com todas elas fora do ciclo FOR-NEXT."
PAULO MACHADO
Vila do Conde

CRIÁMOS ESTA SECÇÃO PARA SI.

COLABORE.

ESCREVA-NOS!

- Inicialmente deve usar uma instrução DIM como, por exemplo, para 20 nomes com 30 caracteres cada:
DIM A$ (20,30)
Pode continuar a usar o ciclo FOR-NEXT para a entrada dos nomes. Quando quiser verificar ou pedir os nomes da lista, pode usar, por exemplo para o 5º nome da lista:

```
PRINT A$ (5)    ou    FOR I=1 TO 20
                      PRINT A$ (I)
                      NEXT I
```

TROCA

O Dr. Nuno Santos (Porto) tem interesse em trocar cassetes de jogos
."ASS/DISASSEMBLER"
."ZX/MONSTER MAZE"
Está especialmente interessado em obter a cassete de XADREZ da PSION.

PROGRAMA : " C U B O "

Autor : Fernando Aguiar /PORTO

Este programa permite o uso do ZX 81 como auxiliar didático, no sentido da visualização de figuras geométricas.

Pode ser alterado no sentido de apresentar outras figuras.

```
   1 REM E£RND)??";" GOSUB TAN
5?? GOSUB ?£RND";" GOSUB TAN 67
090
 270 FAST
 280 LET L=20000
 290 POKE 16388,L-INT (L/256)*25
6
 300 POKE 16389,INT (L/256)
 310 DIM X(5)
 320 DIM Y(5)
 330 DIM O(2)
 340 DIM A(2)
 350 FOR R=0 TO 80 STEP 10
 360 FOR S=1 TO 4
 370 LET T=(R+90*S)*PI/180
 380 LET X(S)=38+(16*COS T)
 390 LET Y(S)=34+(6*SIN T)
 400 NEXT S
 410 IF Y(1)>Y(4) THEN GOTO 480
 420 FOR S=5 TO 2 STEP -1
 430 LET X(S)=X(S-1)
 440 LET Y(S)=Y(S-1)
 450 NEXT S
 460 LET X(1)=X(5)
 470 LET Y(1)=Y(5)
 480 LET O(1)=Y(1)-Y(4)
 490 LET O(2)=Y(1)-Y(2)
 500 LET A(1)=INT (X(4)-X(1))
 510 LET A(2)=INT (X(1)-X(2))
 520 LET T=0
 550 LET T=0
 560 FOR P=0 TO A(1)
 570 LET Q=(O(1)/A(1))*P
 580 FOR S=T TO Q
 590 PLOT X(1)+P,Y(1)-S
 600 PLOT X(2)+P,Y(2)-S
 610 PLOT X(2)+P,Y(2)-S-24
 620 NEXT S
 630 LET T=Q
 640 NEXT P
 650 LET T=0
 660 FOR P=0 TO A(2)
 670 LET Q=(O(2)/A(2))*P
 680 FOR S=T TO Q
 690 PLOT X(2)+P,Y(2)+S
 700 PLOT X(3)+P,Y(3)+S
 710 PLOT X(3)+P,Y(3)+S-24
 720 NEXT S
 730 LET T=Q
 740 NEXT P
 750 FOR P=0 TO 24
 760 FOR S=2 TO 4
 770 PLOT X(S),Y(S)-P
 780 NEXT S
 790 NEXT P
 800 POKE 16518,L-INT (L/256)*25
6
 810 POKE 16519,INT (L/256)
 820 RAND USR 16514
 830 LET L=L+1000
 840 CLS
 850 NEXT R
 860 SLOW
 870 FOR P=20000 TO 28000 STEP 1
000
 880 POKE 16527,P-INT (P/256)*25
6
 890 POKE 16528,INT (P/256)
 900 RAND USR 16526
 910 FOR D=1 TO 5
 920 NEXT D
 930 NEXT P
 940 GOTO 870
 950 SAVE "CUB█"
 960 RUN
```

PROGRAMA " ARRANJOS/PERMUTAÇÕES/COMBINAÇÕES "

AUTOR : HUGO ASSUMPÇÃO - LISBOA NOV 1982

```
  1 REM "ARRANJOS"
  5 LET B$="ARRANJOS"
  9 LET C$="PERMUTACOES"
 10 LET D$="COMBINACOES"
 12 LET E$=" COM "
 14 LET F$=" REPETICAO DE "
 15 LET G$="ELEMENTOS "
 17 LET H$="QUANTOS "
 19 LET I$="TOMADOS "
 21 LET J$="  A "
 23 LET K$="  = "
 25 LET T$=" SEM"
 40 PRINT AT 7,0;"ESCOLHA";AT 1
0,0;"0- STOP ";AT 13,0;"1-";B$;A
T 16,0;"2-";C$;AT 19,0;"3-";D$
 45 INPUT R
 50 IF R=0 THEN STOP
 55 CLS
 60 IF R<1 OR R>3 THEN GOTO 40
 65 PRINT E$(1 TO 4);" OU";T$;F
$,G$;" (C/S)?"
 70 INPUT A$
 75 IF A$<>"S" AND A$<>"C" THEN
 GOTO 70
 80 DIM A(5)
 85 DIM L(5)
 90 PRINT ,,H$;G$;"NO TOTAL?"
 95 INPUT A(1)
100 IF A(1)<=1 THEN GOTO 95
105 LET A(2)=1
110 IF R<>2 THEN PRINT ,,"FORMA
DOS EM CONJUNTOS DE",H$;G$;"?"
115 IF R<>2 THEN INPUT A(2)
120 IF A(2)>=A(1) THEN GOTO 115
125 CLS
130 LET A(3)=A(1)-A(2)
135 LET A(4)=A(2)+A(1)-1
140 LET A(5)=A(1)-1
145 FOR N=1 TO 5
150 LET L(N)=1
155 GOSUB 200
160 NEXT N
165 IF R=1 AND A$="C" THEN PRIN
T B$;E$(1 TO 4);F$;A(1),G$;A(2);
" A ";A(2);" = ";A(1)**A(2)
170 IF R=1 AND A$="S" THEN PRIN
T B$;T$;F$;A(1),G$;I$;A(2);  J$;
A(2);K$;L(1)/L(3)
175 IF R=2 AND A$="C" THEN PRIN
T C$;E$;F$;A(1),G$;K$;A(1)**A(1)
160 IF R=2 AND A$="S" THEN PRIN
T C$;T$;F$;A(1);G$;K$;L(1)
165 IF R=3 AND A$="C" THEN PRIN
T D$;E$;F$;A(1);G$;I$;A(2);J$;A(
2);K$;L(4)/(L(5)*L(2))
190 IF R=3 AND A$="S" THEN PRIN
T D$;T$;F$;A(1),G$;I$;A(2);J$;A(
2);K$;L(1)/(L(2)*L(3))
195 GOTO 40
200 FOR B=1 TO A(N)
205 LET L(N)=L(N)*B
210 NEXT B
215 RETURN
```

Este programa calcula : arranjos; permutações; combinações de elementos de um conjunto.

DEFINIÇÕES :

Arranjos c/ repetição de elementos

$$A^{n}_{m} = n^{m}$$

s/ repetição de elementos

$$A^{n}_{m} = \frac{n!}{(n-m)!}$$

Permutações

caso particular dos arranjos quando $n = m$ c/ repetição :

$$P^{n} = n^{m}$$

s/ repetição

$$P_{n} = n!$$

Combinações

c/ repetição :

$$C^{n}_{m} = \frac{(m+n-1)!}{m!(n-1)!}$$

S/repetição

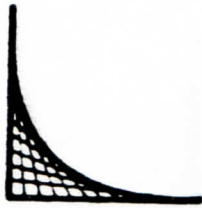
$$C^{n}_{m} = \frac{n!}{m!(n-m)!}$$

A diferença entre arranjos e combinações é a seguinte :

C A B ≠ C B A arranjos

C A B = C B A combinações

NOTA : Este programa trabalha com valores a 17
Rejeita os superiores por overflow na aritmetica

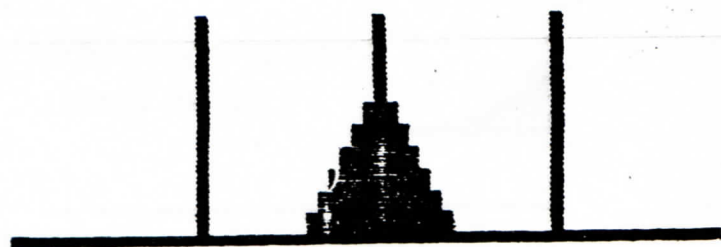
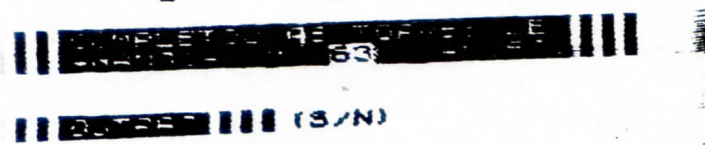
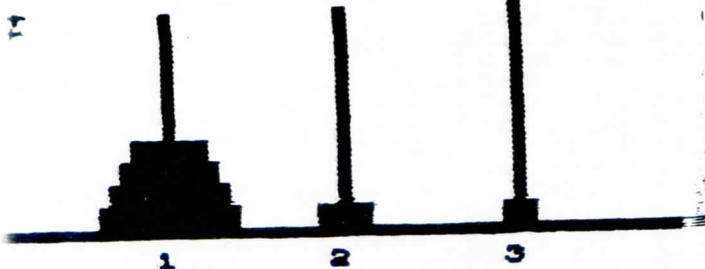
Alteração : 135 LET A(4)=A(2)+A(5)

Programa :

"CONVERSÃO DE BASES"

Autor : FERNANDO PRECES

Lisboa

```
   0 REM PROGRAMA DE ALMEIDA PRE
CES
   5 REM "1"
  10 PRINT AT 3,0;"PROGRAMAS MAT
EMATICOS"
  20 PRINT ,,"CONVERSAO DE BASES
 NUMERICAS."
  25 PRINT ,,,," 1 - CONVERSAO D
E BASE 10 EM:"
  30 PRINT TAB 5;"BASES ( 2 A 9
)"
  35 PRINT ,," 2 - CONVERSAO DE
BASES (2 A 9)"
  40 PRINT TAB 5;"EM BASE 10"
  55 INPUT A
  60 CLS
  62 LET C$="FORA DE ESCALA"
  65 IF A=2 THEN GOTO 245
  75 PRINT ,,"BASE 10 EM BASE (
2 A 9 )"
  80 PRINT ,,,,"INTRODUZA O MODU
LO:"
  85 INPUT A
  87 IF A>9 OR A<2 THEN GOTO 85
  90 PRINT ,,"■BASE PRETENDIDA■
";A
  95 PRINT ,,"INTRODUZA O NUMERO
 DECIMAL:"
 100 INPUT B
 105 PRINT ,,"■NUMERO DECIMAL■ "
;B
 110 LET D=0
 115 LET C=0
 120 LET F=10
 130 LET I=B/A
 135 LET E=INT I
 140 IF E=0 THEN GOTO 190
 145 LET G=B-E*A
 150 IF G=0 THEN GOTO 170
 155 GOSUB 230
 160 LET P=G*Z
 165 LET C=C+P
 170 LET D=D+1
 175 IF D>=F THEN GOTO 210
 180 LET B=E
 185 GOTO 130
 190 GOSUB 230
 195 LET R=B*Z
 200 LET C=C+R
 205 GOTO 215
 210 PRINT ,,C$
 212 LET C=0
 215 PRINT ,,"▙ RESULTADO ▟ ";C
 220 GOTO 485
 230 LET Y=LN 10*D
 235 LET Z=EXP Y
 240 RETURN
 245 PRINT ,,"BASES ( 2 A 9 ) EM
BASE 10"
 250 PRINT ,,"INTRODUZA O MODULO
:"
 255 INPUT A
 257 IF A>9 OR A<2 THEN GOTO 255
 260 PRINT ,,"■ BASE DO NUMERO:
";A
 265 PRINT ,,"INTRODUZA O NUMERO
:"
 270 INPUT B
 275 PRINT ,,"■ NUMERO:   ";B
 280 LET C=0
 285 LET D=0
 290 LET F=10
 295 LET N=B/F
 300 LET E=INT N
 305 IF E=0 THEN GOTO 355
 310 LET G=N-E
 315 LET H=10.001*G
 320 LET M=INT H
 325 LET I=M*(A**D)
 330 LET C=C+I
 335 LET D=D+1
 340 IF D=F THEN GOTO 370
 345 LET B=N
 350 GOTO 295
 355 LET W=INT B
 360 LET P=W*(A**D)
 365 LET C=C+P
 370 PRINT ,,"▙ RESULTADO ▟ ";C
 485 FOR O=1 TO 500
 490 NEXT O
 495 CLS
 499 GOTO 10
```

PROGRAMA
"TORRES DE HANDI" *

```
10 REM
12 REM   "TORRES DE HANOI"
14 REM  SINCLAIR USER,7/82
16 REM
20 CLS
30 PRINT AT 20,0;"
"
35 PRINT AT 21,8;"1";TAB 16;"2
";TAB 24;"3"
40 FOR X=10 TO 19
50 PRINT AT X,5;"       █       █
█"
60 NEXT X
61 DIM A(3,7)
65 DIM A$(7,8)
67 LET A$(1)="
70 LET A$(2)="
80 LET A$(3)="
90 LET A$(4)="
100 LET A$(5)="
110 LET A$(6)="
120 LET A$(7)="
130 FOR Z=7 TO 1 STEP -1
135 LET A(1,Z)=1
140 LET A(2,Z)=Z
145 LET A(3,Z)=1
150 NEXT Z
155 LET C=1
160 FOR Z=1 TO 3
170 FOR Y=7 TO 1 STEP -1
180 PRINT AT Y+12,Z*8-3;A$(A(Z,
Y))
190 NEXT Y
195 NEXT Z
201 PRINT AT 0,23;"LANCE:";C
205 IF A(1,2)=2 OR A(3,2)=2 THE
N GOTO 2000
208 PRINT AT 0,0;"
"
210 PRINT AT 0,0;"DE ?"
220 INPUT J
230 PRINT AT 0,0;J;" PARA ?"
240 INPUT K
250 PRINT AT 0,0;J;" PARA ";K
255 IF J>3 OR J<1 OR K>3 OR K<1
THEN GOTO 1000
260 IF K=J THEN GOTO 1000
270 FOR D=1 TO 7
280 IF A(J,D)=1 THEN GOTO 320
290 LET P=D
300 LET O=A(J,D)
310 GOTO 340
320 NEXT D
330 GOTO 1000
340 FOR D=1 TO 7
350 IF A(K,D)=1 THEN GOTO 370
360 IF A(K,D)<O THEN GOTO 1000
365 IF A(K,D)>1 THEN GOTO 380
370 NEXT D
380 LET D=D-1
390 LET A(K,D)=A(J,P)
400 LET A(J,P)=1
410 LET C=C+1
420 GOTO 160
1000 FOR U=0 TO 20
1010 NEXT U
1020 PRINT AT 0,0;" NAO PODE."
1030 FOR U=0 TO 30
1040 NEXT U
1050 GOTO 208
2000 PRINT AT 0,0;"
 AS TORRES DE
EM ";C-1;" LANCES
2010 PRINT ,,,
2020 PRINT "         (S/N)"
2030 PAUSE 4E3
2040 IF INKEY$="S" THEN RUN
2050 STOP
3000 SAVE "TORRES DE HANOI"
3100 RUN
```

* Devido a problemas na recepção da correspondência, não sabemos quem nos enviou este programa. Agradecemos, pois, que o seu autor nos informe.

```
   1 REM ......................
   2 REM "CIRCUITO"
  12 GOSUB 1000
  14 LET H1=0
  15 LET S=0
  16 LET S1=0
  17 RAND
  19 PRINT "████████████████████████████████"
  20 PRINT "█.............      ............█"
  25 LET Q=0
  30 PRINT "█.███████████      ██████████.█"
  40 PRINT "█.█...........      ..........█.█"
  50 PRINT "█.█.█████████      █████████.█.█"
  60 PRINT "█.█.█.........      ........█.█.█"
  70 PRINT "█.█.█.███████      ███████.█.█.█"
  80 PRINT "█.█.█.█.......      ......█.█.█.█"
  90 PRINT "█.█.█.█.████████████████.█.█.█.█"
 100 FOR A=1TO 4
 110 PRINT "█";TAB 8;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒";TAB 31;"█"
 120 NEXT A
 130 PRINT "█.█.█.█.████████████████.█.█.█.█"
 140 PRINT "█.█.█.█.......      ......█.█.█.█"
 150 PRINT "█.█.█.███████      ███████.█.█.█"
 160 PRINT "█.█.█.........      ........█.█.█"
 170 PRINT "█.█.█████████      █████████.█.█"
 180 PRINT "█.█...........      ..........█.█"
 190 PRINT "█.███████████      ██████████.█"
 200 PRINT "█.............      ............█"
 202 LET V=224
 205 LET G=14
 207 LET H=27
 209 LET V1=V
 210 PRINT "████████████████████████████████"
 215 LET A1=PEEK 16396+256*PEEK 16397
 220 LET B1=0
 225 LET LA=1
 230 LET A=A1+678
 235 LET LB=1
 240 LET B=A1+299
 245 LET A2=0
 250 LET C=1
 252 LET O=A1+INT (RND*660)
 254 IF PEEK O<>27AND PEEK O<>14THEN GOTO 252
 255 IF PEEK O=HTHEN LET V1=V1-1
 256 POKE O,52
 257 LET V=V-1
 258 IF S>1THEN RETURN
 260 LET D=-33
 270 IF A2=HTHEN LET S=S+1
 280 IF PEEK (A+C)=128THEN GOSUB 400
 290 POKE A,G*((A2=H)+(A2=G))
 292 IF S=V1THEN GOSUB 900
 295 IF INKEY$<>""AND PEEK (A+C)=0THEN GOSUB 700
 300 LET A=A+C
 301 IF A=OTHEN LET S1=S1+5
 302 IF A=OTHEN GOSUB 252
 304 LET A2=PEEK A
 305 IF PEEK A=12THEN GOTO 500
 310 POKE A,24
 320 IF PEEK (B+D)=128THEN GOSUB 450
 330 POKE B,B1
 335 IF B1=0AND Q=0AND LA<>LBTHEN GOSUB 800
```

```
337 IF B1<>0THEN LET Q=0
340 LET B=B+D
345 IF PEEK B=24THEN SLOW 500
350 LET B1=PEEK B
360 POKE B,12
370 GOTO 270
400 LET X=0
402 IF C=1THEN LET X=-33
405 IF X=-33THEN GOTO 435
410 IF C=-33THEN LET X=-1
415 IF X=-1THEN GOTO 435
420 IF C=-1THEN LET X=33
425 IF X=33THEN GOTO 435
430 IF C=33THEN LET X=1
435 LET C=X
440 RETURN
450 LET Y=0
452 IF D=-33THEN LET Y=1
455 IF Y=-33THEN GOTO 485
460 IF D=1THEN LET Y=33
465 IF Y=33THEN GOTO 485
470 IF D=33THEN LET Y=-1
475 IF Y=-1THEN GOTO 485
480 IF D=-1THEN LET Y=-33
485 LET D=Y
490 RETURN
500 POKE A,23
510 FOR M=1TO 26
520 RAND USR 16514
530 NEXT M
585 LET S=S+S1
590 SLOW
600 PRINT AT 9,9;"PONTOS ";S
605 IF H1<STHEN LET H1=S
610 PRINT TAB 9;"PONT MAX ";H1
620 PAUSE 400
630 CLS
640 GOTO 15
700 LET A3=A
705 LET A$=INKEY$
710 LET A=A+(((INKEY$="8")-(INKEY$="5"))*(ABS C=33)+((INKEY$="6")-(INKEY$="7
)33*(ABS C=1))*2
720 IF A>A1+726OR A<A1OR PEEK A<>0THEN LET A=A3
730 IF A=A3THEN RETURN
740 LET L5=LA+(C=-1)*(A$="6")+(C=1)*(A$="7")+(C=-33)*(A$="5")+(C=33)*(A$="8"
750 IF L5=LATHEN LET L5=LA-1
755 LET LA=L5
760 RETURN
800 LET Q=1
810 LET D1=D
820 GOSUB 450
830 LET D2=D
840 LET D=D1
850 LET W=LA-LB
860 IF W>1THEN LET W=1
870 IF W<-1THEN LET W=-1
875 LET LB=LB+W
880 LET B=B+W*D2*2
890 RETURN
900 LET S1=S1+S
910 LET S=0
920 LET G=H
930 IF H<>GTHEN GOTO 950
940 LET H=14
950 LET V1=V
```

```
1000 LET M$="042 012 064 006 023 043 035 126 254 118 032 003 016 248 231 198 12
 119 024 242"
1010 FOR M=16514TO 16533
1020 POKE M,VAL M$(TO 3)
1030 LET M$=M$(5TO )
1040 NEXT M
1050 RETURN
```

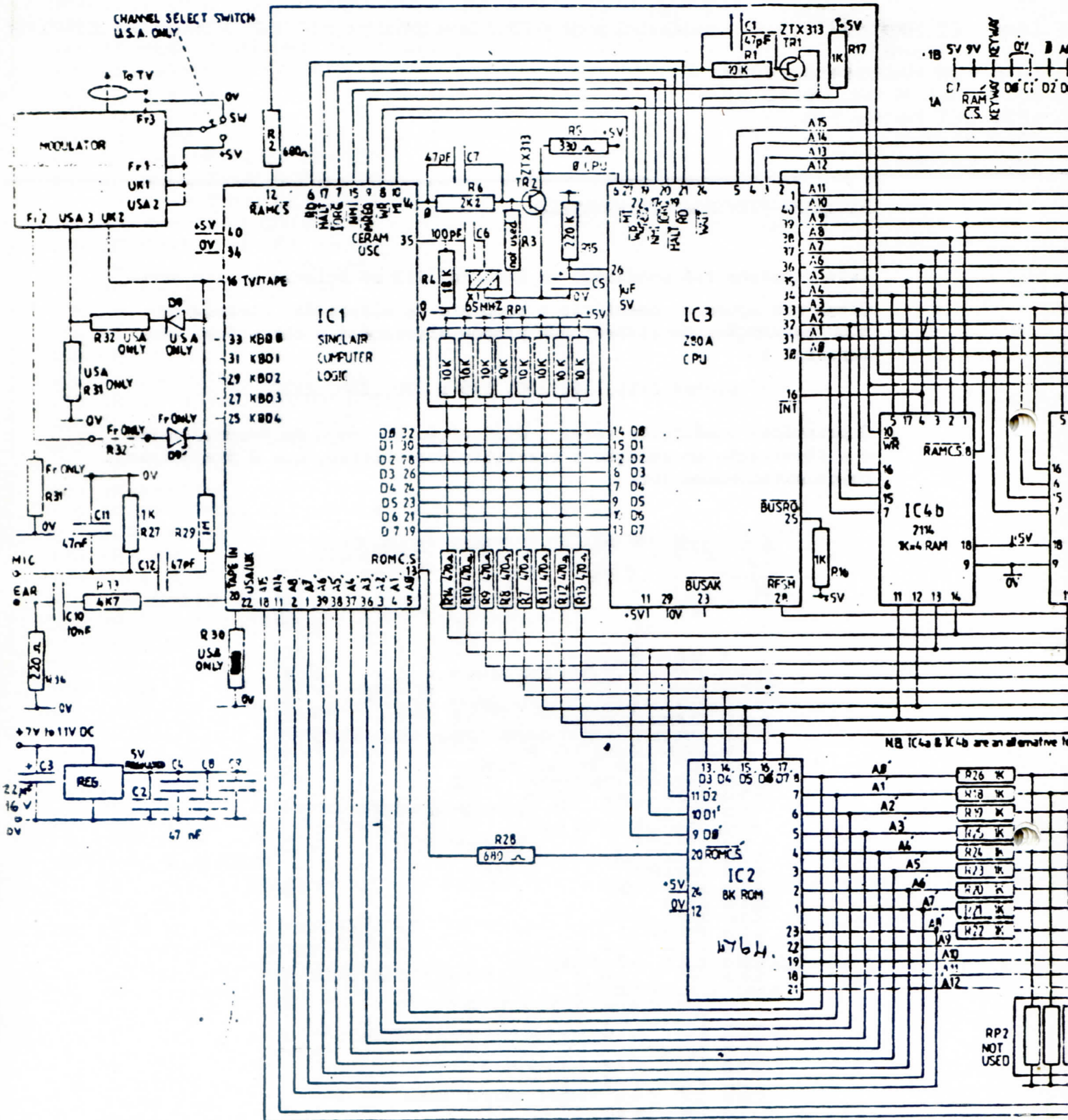
PROGRAMA "INVERSAO DE MATRIZES "

Este programa foi publicado em Setembro/1982 no boletim numero zero.

Chega-nos agora o comentário e proposta de alteração, através do HUGO Assumpção; de Lisboa; que propõe alterações e correções, em especial :

Linhas 116; 118; 140; 145 ; 230, 250; 273

O principal comentário versa a diminuição do tempo de processamento e a observação em relação à exactidão dos valores, que é prejudicada pelos arredondamentos.

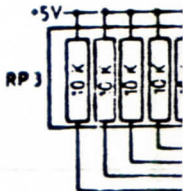
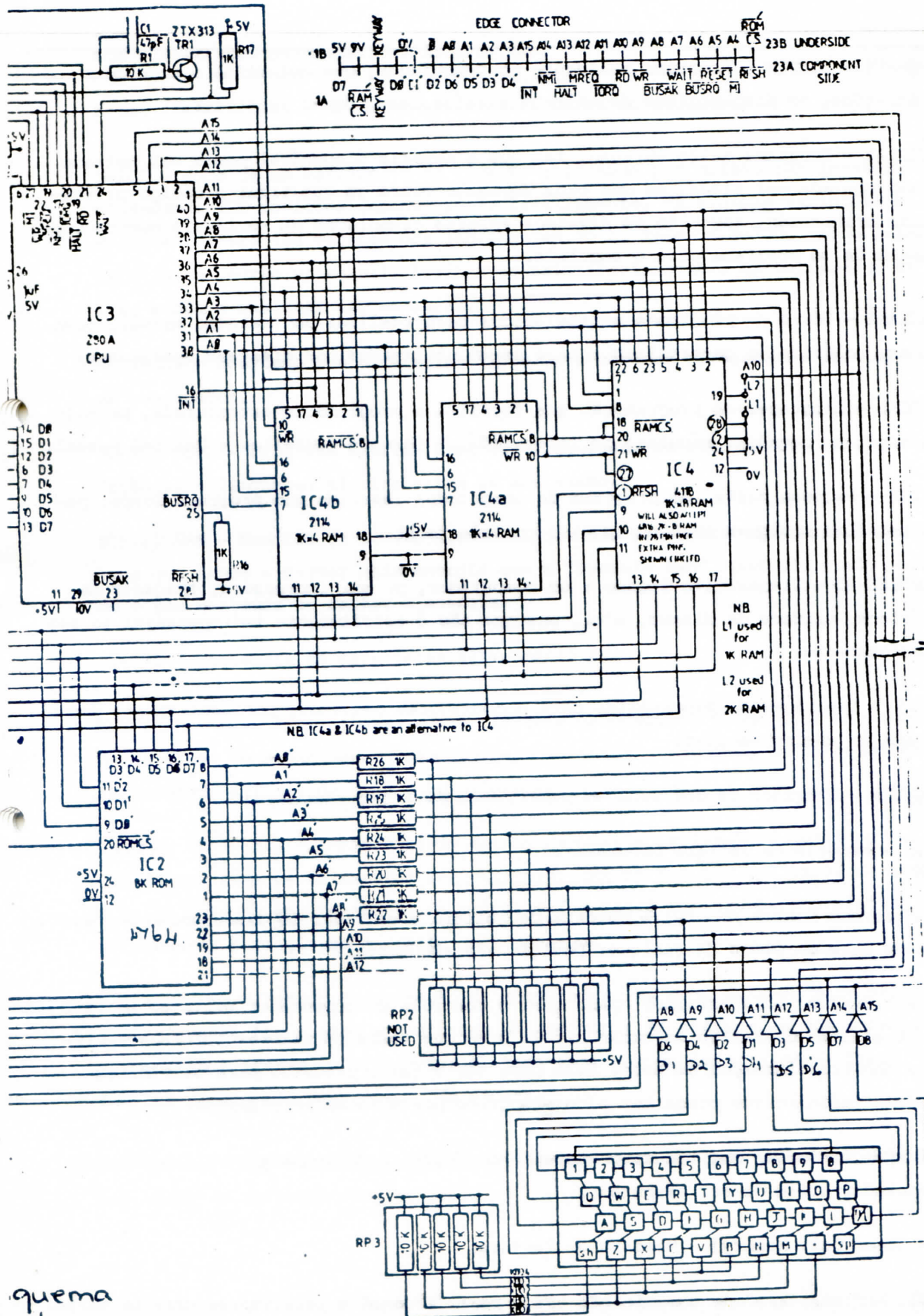
```
   1 REM "M/I"
  20 PRINT "ORDEM N= ";
  30 INPUT N
  35 PRINT N
  40 DIM A(N,N)
  45 PRINT "ENTRADA DOS VALORES"
  50 FOR I=1 TO N
  55 IF I=6 THEN CLS
  60 PRINT "LINHA ";I
  70 FOR J=1 TO N
  80 PRINT " ";"COLUNA ";J;
  90 INPUT A(I,J)
  95 PRINT " = ";A(I,J);
 100 NEXT J
 105 PRINT
 110 NEXT I
 115 CLS
 116 LET B=1
 118 FAST
 120 FOR X=1 TO N
 130 LET DI=A(X,1)
 140 IF DI=0 THEN PRINT "MATRIZ
NAO INVERTIVEL"
 145 IF DI=0 THEN STOP
 150 FOR Y=1 TO N-1
 160 LET A(X,Y)=A(X,Y+1)/DI
 170 NEXT Y
 180 LET A(X,N)=1/DI
 190 FOR Z=1 TO N
 200 IF Z=X THEN GOTO 260
 210 LET A(Z,1)=0
 220 FOR Y=1 TO N-1
 230 LET A(Z,Y)=A(Z,Y+1)-B*A(X,Y
)
 240 NEXT Y
 250 LET A(Z,N)=-B*A(X,N)
 260 NEXT Z
 270 NEXT X
 273 SLOW
 275 PRINT "        "
 280 FOR I=1 TO N
 290 FOR J=1 TO N
 300 PRINT A(I,J);"  "
 310 NEXT J
 320 PRINT
 330 NEXT I
 340 PRINT
 350 PRINT "   "
```

DRW No. SRC 049

SINCLAIR ZX81

14/5/82

Esquema Electrico do ZX81

quema
ectrico
do ZX 81

0276 66104.

OPERATING SYSTEM - Programa residente no computador, que determina o modo como as instruções, os dispositivos de entrada e saída, etc. devem funcionar.

OVERLAY - Programa muito extenso, tal que a memória disponível possa dar entrada e ser operada por fases, em que cada segmento cobre ou substitui o código previamente armazenado, enquanto os diversos valores acumulados em variáveis comuns continuam de um programa para o seguinte.

PACKAGE - Série de programas estabelecidos para realizar um trabalho normal, como p. ex. salários, e generalizados para serem utilizados por diversos operadores.

PASCAL - Linguagem de programação que facilita a programação estruturada, principalmente em pequenas máquinas de interacção. O termo é posterior a Gabriel Pascal.

PATCH - Pequena parte do programa do computador inserido num programa longo, para remediar qualquer erro ou defeito que ele tenha.

PERIPHERAL - Dispositivo ligado a um computador, p. ex. impressora, traçador de gráficos, unidade de discos, etc., mas que não é estritamente indispensável ao seu funcionamento.

PILOT - Linguagem de programação para computadores pequenos, particularmente destinada ao ensino escolar.

PLOTTER - Traçador de gráficos em papel, comandado por computador.

PROM - Memória de leitura programável.

RAM (Random Access Memory) - Memória de leitura e escrita. Os dados podem ser escritos para ou lidos de qualquer posição não sequencial.

RESET (Button) - Interruptor pelo qual o controle do computador volta ao monitor ou ao sistema operativo de baixo nível, e em que todos os valores internos variáveis são transformados em zero. Este pode ser o único processo para sair de um círculo infinito que possa ter sido executado por erro de programação.

RETURN - Instrução que remete o conteúdo do "buffer" do teclado para a memória do computador a fim de ser executado.

ROM (Read Only Memory) - Memória de leitura.

RS232 - "Interface" de comunicação usado para "modems" e para impressoras em série.

RUN - Instrução para executar um programa.

N-SEC (Nanosecond) - Milésima-milionésima parte do segundo.

S-100 - Nome de um barramento ou ligação utilizado por muitos fabricantes, e compreendendo 100 posições (condutores). Infelizmente, há algumas variações mínimas entre as diferentes versões dos fabricantes do barramento S-100, mas o I-EEE já definiu um padrão universal. Destina-se prioritariamente a um barramento de memória, e não a qualquer uso em geral.

SOFTWARE - Conjunto dos diferentes tipos de programas necessários para operar com um computador.

SOURCE CODE - Programa escrito numa das linguagens de alto nível, exigindo compilação para linguagem máquina antes de ser usado.

STATIC RAM - Memória de acesso aleatório que não necessita realimentação contínua, mas que tende a gastar mais energia que a "Dynamic RAM", podendo ainda perder todo o seu conteúdo se falhar a energia.

STRING - Sequencia de caracteres alfanuméricos.

TERMINAL - Dispositivo, geralmente afastado do computador, no qual os dados podem dar entrada ou saída de um trabalho de comunicação - p. ex., um tele-impressor funcionando através de linhas telefónicas.

THERMAL (Printer) - Impressora de matriz na qual a impressão é feita por aquecimento de uma rede de filamentos que constitui uma matriz (5×7 p. ex.), de modo que o calor leva ao escurecimento de pontos de um papel tratado para o efeito, com a consequente formação do caracter seleccionado.

TIME-SHARING - Método de operação de computador, no qual dois ou mais utilizadores aparentemente tem acesso e controlam uma máquina. Na prática, o que acontece é que o computador atende um utilizador de cada vez, mas em intervalos de tempo tão curtos que parece não haver tempo de espera.

VDU (Visual Display Unit) - Visor do tipo televisão no qual podem ser exibidas as mensagens de um computador (video).

WORD - Número de bits que um computador necessita para processar um grupo de informação - p. ex. palavras 16-bit. As mais vulgarizadas são as palavras de 8-bit que são chamadas 1 byte.

WORD PROCESSOR - Computador cujo "software" permite a entrada, a cópia, o armazenamento, a formatação e a impressão de textos em vez do processamento numérico.

(Fim)

L I N G U A G E M   M Á Q U I N A

- Continuação do artigo apresentado no nº anterior (pág. 6 - 9, bol.2)

Observe o código máquina do programa 2 (pág. 9, boletim 2) e os dados introduzidos, verificando como entrar com o código máquina para uma instrução REM. Depois faça: RUN
Isso originará um número que é a adição dos códigos dos caracteres em Rem 2. Experimente editar Rem 2 e introduza diferentes caracteres. Dê entrada ao programa 3 (pág. 9, boletim 2) usando a mesma técnica. Faça Run 800 e escreva o código máquina. Este programa subtrai a primeira variável de Rem 2 da segunda variável.
Em ambos os casos, só serão correctas as respostas positivas entre Ø e 255. Tente novamente mudar as variáveis em Rem 2 e observe o efeito. Introduza o programa 4 (pág. 9, boletim 2).
Este multiplica os 2 códigos dos caracteres em Rem 2 conjuntamente. Mais uma vez, só respostas positivas entre Ø e 255 serão correctas. A resposta está contida na variável simples C. Modifique o programa 1 para programa 1a. Agora, introduza o programa 5.*
É necessário criar um ficheiro de exibição ("display file") com um programa Basic antes de introduzir nele caracteres alternativos com uma rotina em código-máquina.
Não esqueça, ao passar de uma linha para outra, que todas as linhas, mesmo as vazias, terminam com o caracter NewLine (pág. 178, manual do Sinclair) e que "Print At" começa na coluna 0 (pág. 129, manual do Sinclair).

Utilizando o programa BASIC, os asteriscos vão sendo exibidos no écran, ao passo que o quadro do código máquina é quase instantâneo. Experimente colocar uma pausa no BASIC para demontrar isto mesmo. Experimente também diferentes padrões no écran, fazendo variar partes do código máquina. Se o conseguir, o mais difícil está ultrapassado.

→

* V. programa 5 nas págs. seguintes

Endereço	Código Máquina	Mnemónica	Basic
16514	33 12 64	LD HL NN	LET HL = 16396
	94	LD E(HL)	LET E = PEEK HL
	35	INC HL	LET HL = HL + 1
	86	LD D(HL)	LET D = PEEK HL
	33 3 0	LD HL NN	LET HL = 3
	25	ADD HL DE	LET HL = HL + DE
	125	LD A L	LET A = L Guarda a po
	79	LD C A	LET C = A sição inicial
	124	LD A H	LET A = H
	71	LD B A	LET B = A
16528	54 135	LD(HL)N	POKE HL, 135 Linha trans
	35	INC HL	LET HL = HL + 1 versal
	54 131	LD(HL)N	POKE HL, 131
	35	INC HL	LET HL = HL + 1
	54 131	LD(HL)N	POKE HL, 131
	35	INC HL	LET HL = HL + 1
	54 131	LD(HL)N	POKE HL, 131
	35	INC HL	LET HL = HL + 1
	54 131	LD(HL)N	POKE HL, 131
	35	INC HL	LET HL = HL + 1
	54 131	LD(HL)N	POKE HL, 131
	35	INC HL	LET HL = HL + 1
	54 131	LD(HL)N	POKE HL, 131
	35	INC HL	LET HL = HL + 1
	54 131	LD(HL)N	POKE HL, 131
	35	INC HL	LET HL = HL + 1
	54 4	LD(HL)N	POKE HL, 4
16555	17 10 0	LD DE NN	LET DE = 10 Linha descen
	25	ADD HL DE	LET HL = HL + DE dente
	54 5	LD(HL)N	POKE HL, 5 da direita
	25	ADD HL DE	LET HL = HL + DE
	54 5	LD(HL)N	POKE HL, 5
	25	ADD HL DE	LET HL = HL + DE
	54 5	LD(HL)N	POKE HL, 5
	25	ADD HL DE	LET HL = HL + DE
	54 5	LD(HL)N	POKE HL, 5
	25	ADD HL DE	LET HL = HL + DE
	54 1	LD(HL)N	POKE HL, 1
16572	121	LD A C	LET A = C Lê a posi-
	111	LD L A	LET L = A ção inicial
	120	LD A B	LET A = B
	103	LD H A	LET H = A
	25	ADD HL DE	LET HL = HL + DE Linha
	54 133	LD(HL)N	POKE HL, 133 descendente
	25	ADD HL DE	LET HL = HL + DE esquerda
	54 133	LD(HL)N	POKE HL, 133
	25	ADD HL DE	LET HL = HL + DE
	54 133	LD(HL)N	POKE HL, 133
	25	ADD HL DE	LET HL = HL + DE
	54 133	LD(HL)N	POKE HL, 133
	25	ADD HL DE	LET HL = HL + DE
	54 2	LD(HL)N	POKE HL, 2
16591	35	INC HL	LET HL = HL + 1 Linha
	54 3	LD(HL)N	POKE HL, 3 transversal
	35	INC HL	LET HL = HL + 1 do fundo

Programa 5

→

Endereço	Código Máquina	Mnemónica	Basic
	54 3	LD(HL)N	POKE HL, 3
	35	INC HL	LET HL = HL + 1
	54 3	LD(HL)N	POKE HL, 3
	35	INC HL	LET HL = HL + 1
	54 10	LD(HL)N	POKE HL, 10
	35	INC HL	LET HL = HL + 1
	54 3	LD(HL)N	POKE HL, 3
	35	INC HL	LET HL = HL + 1
	54 3	LD(HL)N	POKE HL, 3
	35	INC HL	LET HL = HL + 1
	54 3	LD(HL)N	POKE HL, 3
	201	RET	

Programa 5 (continuação)

```
   1 REM 5£RND???5  ; ????Q 7Q 7Q
 7Q 7Q 7Q 7Q 7Q 7Q )    ;Q ; Q ; Q ;
Q ;Q ????;Q ;Q ;Q ; Q   ;Q 7Q 7Q 7Q
 7Q 7Q 7Q 7Q TAN Ø 123 4 5678 9Ø1 234
5678901234567890 12 345 6 789Ø 123 456
789Ø
 2ØØ CLS
 21Ø SLOW
 22Ø FOR K = 2 TO 7
 23Ø PRINT AT K,8;"  "
 24Ø NEXT K
 25Ø LET C = USR 16514
 3ØØ STOP
 8ØØ FAST
 8Ø1 FOR K = 16514 TO 16664
 81Ø SCROLL
 82Ø INPUT J
 83Ø POKE K,J
 84Ø PRINT AT 7,Ø;K;TAB 8;J
 85Ø NEXT K
```

Programa 5 e um exemplo de
exibição do quadro

CLUBE Z-8Ø - QUANTOS SOMOS? ONDE ESTAMOS?

Até ao momento, contamos com 105 associados dispersos por vários locais do país. O quadro seguinte apresenta essa distribuição (as zonas foram determinadas na base dos códigos postais).

Local	Nº	Local	Nº
ÁGUEDA	1	NINE	1
ALMADA	1	OLIVEIRA DE FRADES	1
AVEIRO	1	PENICHE	2
BARREIRO	1	PORTALEGRE	1
BRAGA	3	PORTO	38
CALDAS DA RAINHA	2	PÓVOA DE VARZIM	1
CASCAIS	1	QUARTEIRA	1
COIMBRA	1	RIO TINTO	2
COVILHÃ	1	SACAVÉM	2
ERMESINDE	1	SETÚBAL	2
ÉVORA	4	SINTRA	1
FELGUEIRAS	1	TORRES VEDRAS	1
FUNCHAL	1	TROFA	1
FUNDÃO	1	VAGOS	1
GONDOMAR	5	VALE DE CAMBRA	1
LEIRIA	1	VIANA DO CASTELO	1
LISBOA	11	VILA NOVA DE GAIA	4
MAIA	1	VILA REAL DE STO. ANTÓNIO	1
MATOSINHOS	3	VILA VELHA DE RÓDÃO	1
MONTE DE CAPARICA	1		

O CLUBE Z-8Ø lembra a todos que a ele se associaram através de prestação trimestral que esta terminou no final de Dezembro.
Assim, se deseja continuar a fazer parte do CLUBE Z-8Ø, solicitamos-lhe que renove a sua inscrição utilizando o cupão anexo.

CLUBE Z-8Ø - RENOVAÇÃO DE INSCRIÇÃO

Desejo renovar a minha inscrição no CLUBE Z-8Ø, com mais uma prestação de

- ☐ *375$00 (Jan., Fev. e Março)
- ☐ *750$00 (Jan. a Junho)
- ☐ *1 125$00 (Jan. a Setembro)

Envio:

☐ *Cheque nº ____________ ☐ *Vale Postal nº ____________ ☐ *Dinheiro

NOME: ______________________________ DATA: __________

ENDEREÇO ______________________________